Signals and Circuits

ENGR 35500

Boolean Representation and Theorems

Text books:

Ulaby, Fawwaz T., and Maharbiz, Michael M., *Circuits*, 2nd Edition, National Technology and Science Press, 2013.
Floyd, T. L., and Buchla, D. M., *Electroics Fundamentals: Circuits, Devices & Applications*, 8th Edition, Pearson, 2009.
Digital Electronics: Principles, Devices and Applications. Anil K. Maini , John Wiley & Sons, 2007.

KENT STATE UNIVERSITY

Boolean algebra

- Boolean algebra is mathematics of logic

- When we learned numbers like 1, 2, 3, we also then learned how to add, multiply, etc. with them. Boolean Algebra covers operations that we can do with 0's and 1's. Computers do these operations ALL THE TIME and they are basic building blocks of computation inside your computer program.

Boolean algebra vs ordinary algebra

- In ordinary algebra, the letter symbols can take on any number of values including infinity. In Boolean algebra, they can take on either of two values, that is, 0 and 1.
- The values assigned to a variable have a numerical significance in ordinary algebra, whereas in its Boolean counterpart they have a logical significance.
- While '.' and '+' are respectively the signs of multiplication and addition in ordinary algebra, in Boolean algebra '.' means an AND operation and '+' means an OR operation. For instance, A+B in ordinary algebra is read as A plus B, while the same in Boolean algebra is read as A OR B.
- Boolean algebra captures the essential properties of both logic operations such as AND, OR and NOT and set operations such as intersection, union and complement.
- Boolean algebra may also be defined to be a set A supplied with two binary operations of logical AND, logical OR, a unary operation of logical NOT and two elements, namely logical FALSE (0) and logical TRUE (1).

Boolean algebra

- Variables are the different symbols in a Boolean expression;
- Each occurrence of a variable or its complement is called a literal;
- A term is the expression formed by literals and operations at one level.

E. g.

$$\overline{A} + A.B + A.\overline{C} + \overline{A}.B.C$$

- A, B, C three variables;
- Eight literals
- Five terms including four AND terms and the OR term that combines the first-level AND terms

Boolean algebra

➢ Equivalent

- Two given Boolean expressions are said to be equivalent if one of them equals ‘1’ only when the other equals ‘1’ and also one equals ‘0’ only when the other equals ‘0’.

➢ Complement of each other

- Two Boolean expressions are said to be the complement of each other if one expression equals ‘1’ only when the other equals ‘0’, and vice versa.
- The complement of a given Boolean expression is obtained by complementing each literal, changing all ‘.’ to ‘+’ and all ‘+’ to ‘.’, all 0s to 1s and all 1s to 0s

E. g.

Given Boolean expression

$$\overline{A}.B + A.\overline{B}$$

Corresponding complement

$$(A + \overline{B}).(\overline{A} + B)$$

‘ is NOT/complement

➢ Precedence

NOT > AND > OR

Theorems of Boolean algebra

- Theorem 1

(Operations with '0' and '1')

$$\text{(a) } 0.X = 0 \quad \text{and} \quad \text{(b) } 1 + X = 1$$

- Theorem 2

(Operations with '0' and '1')

$$\text{(a) } 1.X = X \quad \text{and} \quad \text{(b) } 0 + X = X$$

- Theorem 3

(Idempotent or Identity Laws)

$$\text{(a) } X.X.X.\ldots.X = X \quad \text{and} \quad \text{(b)} X + X + X + \cdots + X = X$$

E. g.

$$(A.\overline{B}.\overline{B} + C.C).(A.\overline{B}.\overline{B} + A.\overline{B} + C.C) = (A.\overline{B} + C).(A.\overline{B} + A.\overline{B} + C)$$
$$= (A.\overline{B} + C).(A.\overline{B} + C) = A.\overline{B} + C$$

Theorems of Boolean algebra

- Theorem 4

(Complementation Law)

(a) $X.\overline{X} = 0$ and (b) $X + \overline{X} = 1$

E. g.

$$(A + B.C)(\overline{A + B.C}) = 0 \quad \text{and} \quad (A + B.C) + (\overline{A + B.C}) = 1$$

- Theorem 5

(Commutative Laws)

(a) $X + Y = Y + X$ and (b) $X.Y = Y.X$

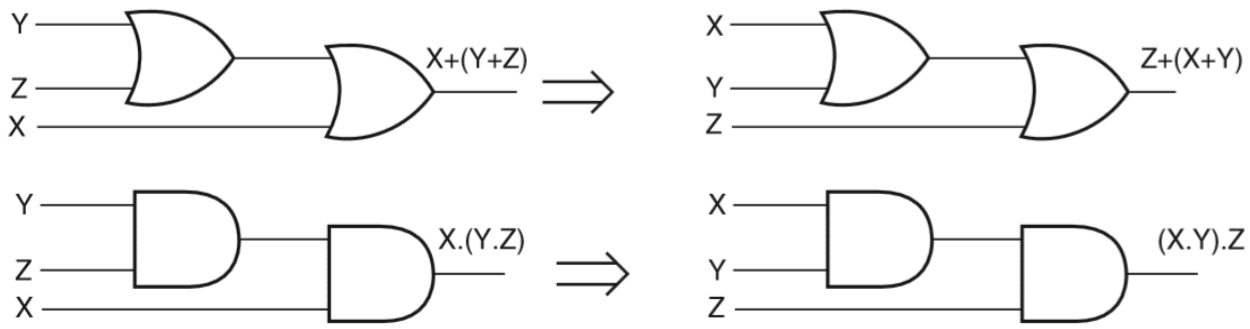
- Theorem 6

(Associative Laws)

(a) $X + (Y + Z) = Y + (Z + X) = Z + (X + Y)$

(b) $X.(Y.Z) = Y.(Z.X) = Z.(X.Y)$

E. g.



KENT STATE UNIVERSITY

Theorems of Boolean algebra

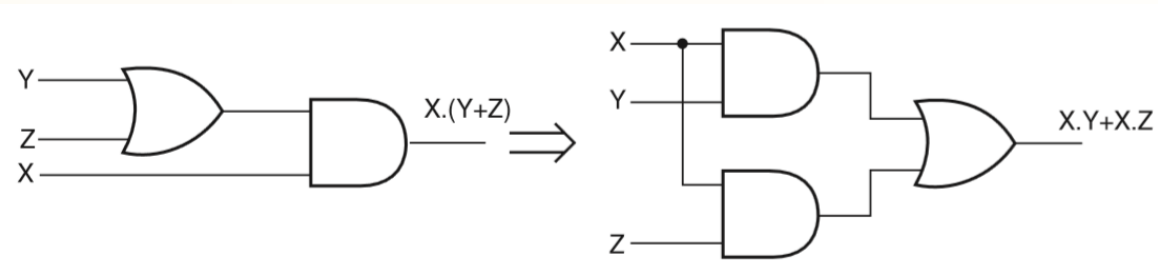
- Theorem 7

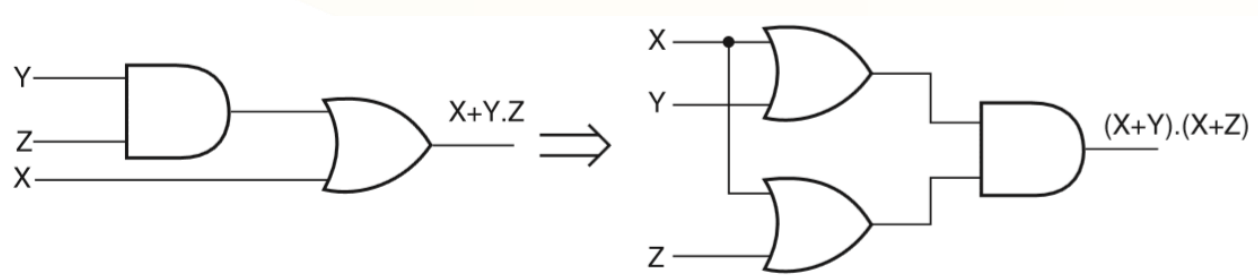
(Distributive Laws)

(a) $X.(Y+Z) = X.Y + X.Z$ and (b) $X + Y.Z = (X+Y).(X+Z)$

E. g.

$$\overline{A}.\overline{B} + \overline{A}.B + A.\overline{B} + A.B = \overline{A}.(\overline{B}+B) + A.(\overline{B}+B) = \overline{A}.1 + A.1 = \overline{A} + A = 1$$





- Theorem 8

(a) $X.Y + X.\overline{Y} = X$ and (b) $(X+Y).(X+\overline{Y}) = X$

E. g.

$X.Y + X.\overline{Y} = X.(Y+\overline{Y}) = X.1 = X$ and $(X+Y).(X+\overline{Y}) = X + Y.\overline{Y} = X + 0 = X$

KENT STATE UNIVERSITY

Theorems of Boolean algebra

- Theorem 9

$$\text{(a) } (X+\overline{Y}).Y = X.Y \quad \text{and} \quad \text{(b) } X.\overline{Y}+Y = X+Y$$

E. g.

$$(X+\overline{Y}).Y = X.Y+\overline{Y}.Y = X.Y$$

- Theorem 10 *(Absorption Law or Redundancy Law)*

$$\text{(a) } X+X.Y = X \quad \text{and} \quad \text{(b) } X.(X+Y) = X$$

E. g.

$$X+X.Y = X.(1+Y) = X.1 = X$$

- Theorem 11

$$\text{(a) } Z.X+Z.\overline{X}.Y = Z.X+Z.Y$$

$$\text{(b) } (Z+X).(Z+\overline{X}+Y) = (Z+X).(Z+Y)$$

E. g.

$$(A+\overline{B}).(\overline{A}+\overline{B}+C).(\overline{A}+\overline{B}+D)$$

$$= (A+\overline{B}).(\overline{B}+C).(\overline{A}+\overline{B}+D) = (A+\overline{B}).(\overline{B}+C).(\overline{B}+D)$$

Theorems of Boolean algebra

- Theorem 12

(Consensus Theorem)

(a) $X.Y + \overline{X}.Z + Y.Z = X.Y + \overline{X}.Z$

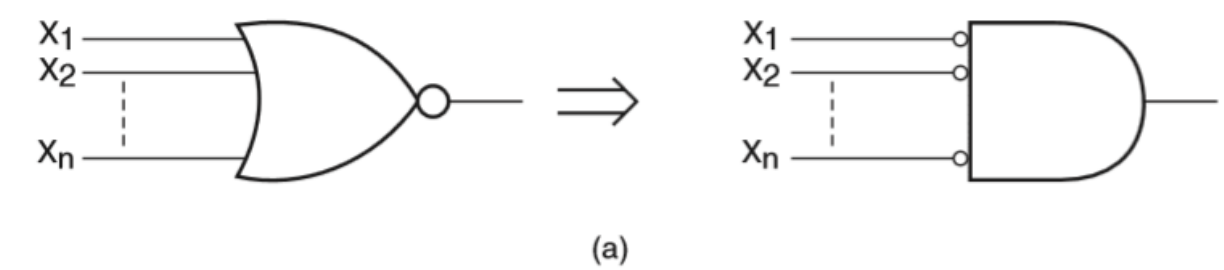
(b) $(X + Y).(\overline{X} + Z).(Y + Z) = (X + Y).(\overline{X} + Z)$

- Theorem 13

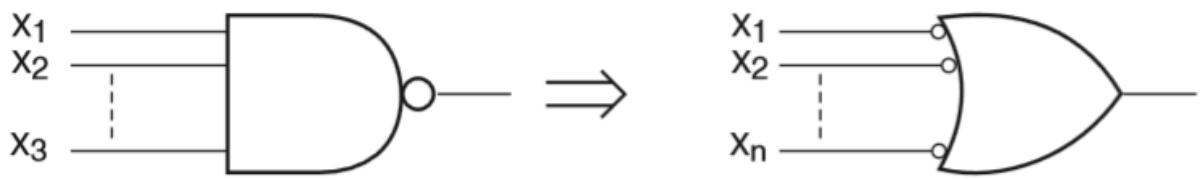
(DeMorgan's Theorem)

(a) $[\overline{X_1 + X_2 + X_3 + \ldots + X_n}] = \overline{X_1}.\overline{X_2}.\overline{X_3}.\ldots.\overline{X_n}$

(b) $[X_1.X_2.X_3.\ldots.X_n] = [\overline{X_1} + \overline{X_2} + \overline{X_3} + \ldots + \overline{X_n}]$



(a)



(b)

KENT STATE UNIVERSITY

Theorems of Boolean algebra

- Theorem 14 *(Transposition Theorem)*

$$\text{(a) } X.Y + \overline{X}.Z = (X + Z).(\overline{X} + Y)$$

$$\text{(b) } (X + Y).(\overline{X} + Z) = X.Z + \overline{X}.Y$$

- Theorem 15

$$\text{(a) } X.f(X, \overline{X}, Y, Z, \ldots) = X.f(1, 0, Y, Z, \ldots)$$

$$\text{(b) } X + f(X, \overline{X}, Y, Z, \ldots) = X + f(0, 1, Y, Z, \ldots)$$

- Theorem 16

$$\text{(a) } f(X, \overline{X}, Y, \ldots, Z) = X.f(1, 0, Y, \ldots, Z) + \overline{X}.f(0, 1, Y, \ldots, Z)$$

$$\text{(b) } f(X, \overline{X}, Y, \ldots, Z) = [X + f(0, 1, Y, \ldots, Z)][\overline{X} + f(1, 0, Y, \ldots, Z)]$$

Theorems of Boolean algebra

- Theorem 1 *(Operations with '0' and '1')*

  (a) $0.X = 0$ and (b) $1 + X = 1$

- Theorem 2 *(Operations with '0' and '1')*

  (a) $1.X = X$ and (b) $0 + X = X$

- Theorem 3 *(Idempotent or Identity Laws)*

  (a) $X.X.X.\ldots.X = X$ and (b) $X + X + X + \cdots + X = X$

- Theorem 4 *(Complementation Law)*

  (a) $X.\overline{X} = 0$ and (b) $X + \overline{X} = 1$

- Theorem 5 *(Commutative Laws)*

  (a) $X + Y = Y + X$ and (b) $X.Y = Y.X$

- Theorem 6 *(Associative Laws)*

  (a) $X + (Y + Z) = Y + (Z + X) = Z + (X + Y)$ (b) $X.(Y.Z) = Y.(Z.X) = Z.(X.Y)$

- Theorem 7 *(Distributive Laws)*

  (a) $X.(Y + Z) = X.Y + X.Z$ and (b) $X + Y.Z = (X + Y).(X + Z)$

- Theorem 8

  (a) $X.Y + X.\overline{Y} = X$ and (b) $(X + Y).(X + \overline{Y}) = X$

- Theorem 9

  (a) $(X + \overline{Y}).Y = X.Y$ and (b) $X.\overline{Y} + Y = X + Y$

- Theorem 10 *(Absorption Law or Redundancy Law)*

  (a) $X + X.Y = X$ and (b) $X.(X + Y) = X$

- Theorem 11

  (a) $Z.X + Z.\overline{X}.Y = Z.X + Z.Y$

  (b) $(Z + X).(Z + \overline{X} + Y) = (Z + X).(Z + Y)$

- Theorem 12 *(Consensus Theorem)*

  (a) $X.Y + \overline{X}.Z + Y.Z = X.Y + \overline{X}.Z$ (b) $(X + Y).(\overline{X} + Z).(Y + Z) = (X + Y).(\overline{X} + Z)$

- Theorem 13 *(DeMorgan's Theorem)*

  (a) $[\overline{X_1 + X_2 + X_3 + \ldots + X_n}] = \overline{X_1}.\overline{X_2}.\overline{X_3}.\ldots.\overline{X_n}$

  (b) $[\overline{X_1.X_2.X_3.\ldots.X_n}] = [\overline{X_1} + \overline{X_2} + \overline{X_3} + \ldots + \overline{X_n}]$

- Theorem 14 *(Transposition Theorem)*

  (a) $X.Y + \overline{X}.Z = (X + Z).(\overline{X} + Y)$ (b) $(X + Y).(\overline{X} + Z) = X.Z + \overline{X}.Y$

Practice

Simply the following Boolean expressions to a minimum number of literals (' is NOT/complement)

$$xy + xy'$$

Or $xy + x\bar{y}$

$$(A + B)'(A' + B')'$$

Or $\overline{A + B}\ \overline{\bar{A} + \bar{B}}$

KENT STATE UNIVERSITY

Signals and Circuits

ENGR 35500

Sequential Circuits

Text books:

Ulaby, Fawwaz T., and Maharbiz, Michael M., *Circuits*, 2nd Edition, National Technology and Science Press, 2013.

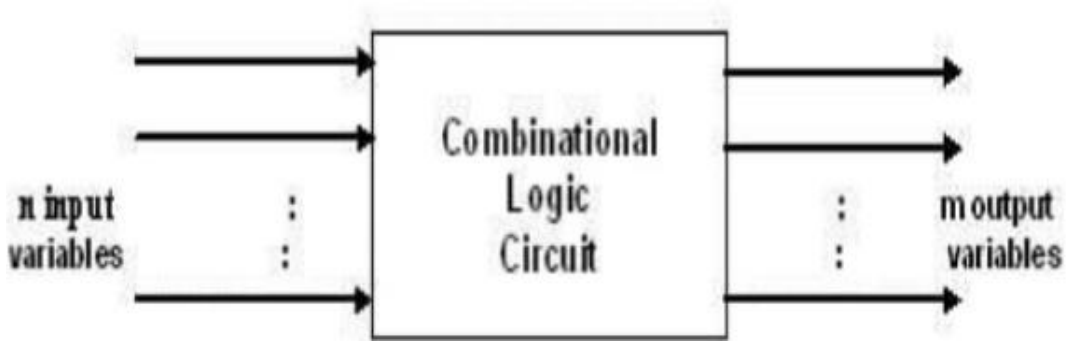
Floyd, T. L., and Buchla, D. M., *Electroics Fundamentals: Circuits, Devices & Applications*, 8th Edition, Pearson, 2009.

Anil K. Maini, Digital Electroics: Principles, Devices and Applications

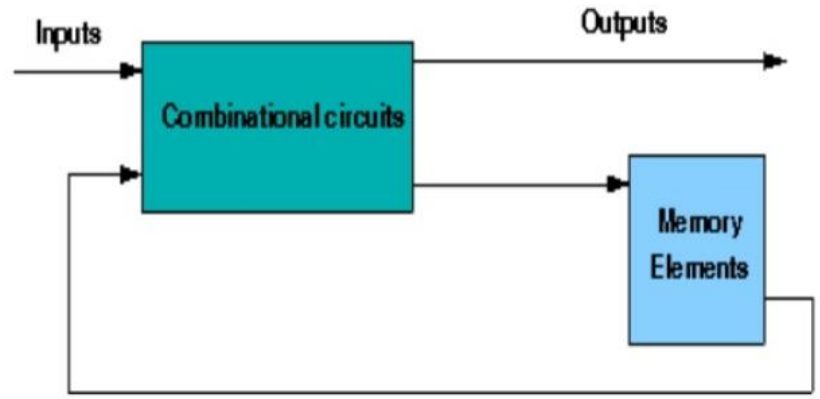
KENT STATE UNIVERSITY

Sequential Circuits

- Most digital systems like digital watches, digital phones, digital computers, digital traffic light controllers and so on require memory elements
- Memory elements are digital circuits that can store and retrieve data in the form of 1's and 0's.
- The output of the systems with memory depends not only on present inputs but also on what has happened in the past
- A **latch** is a temporary storage device that has two stable states (bistable). It is a basic form of memory.
- SR latch is an example of memory circuits that can store one bit of information



Combinational circuit

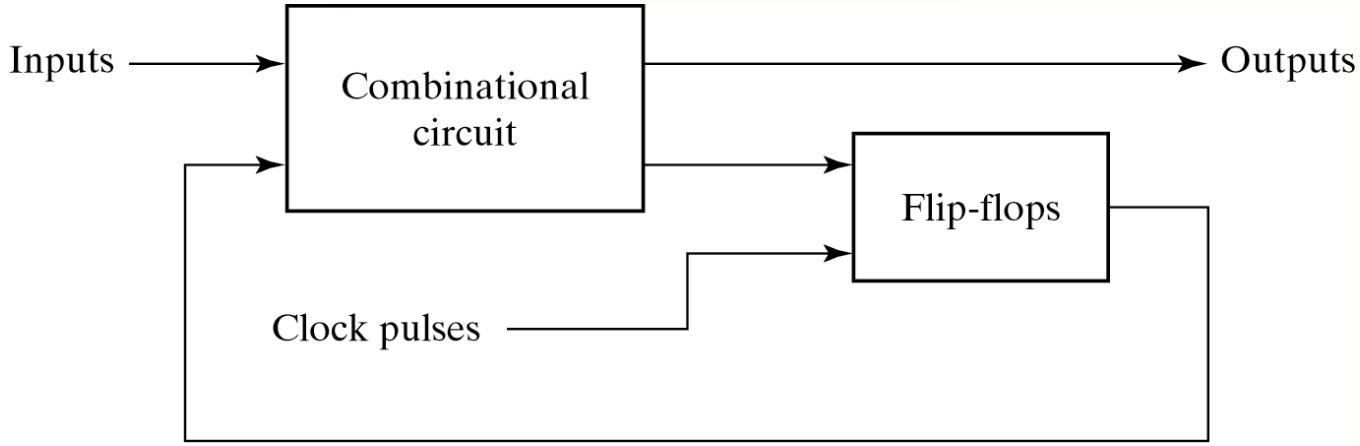


sequential circuit

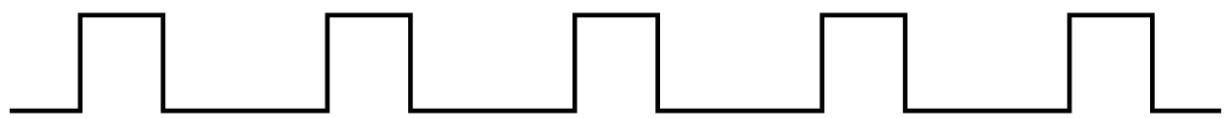
KENT STATE UNIVERSITY

Synchronous vs. Asynchronous

- The behavior of a synchronous sequential circuit depends upon the any input signal at any instant of time and order of input change. This synchronization is achieved by clock generators that provides clock pulses (The storage elements used in clocked sequential circuits are called *flip-flops*.
- In asynchronous sequential circuits the storage elements are time delay devices (i.e., storage is because of their propagation delay). They implemented by feedback that may cause instability in Asynchronous circuits.



(a) Block diagram

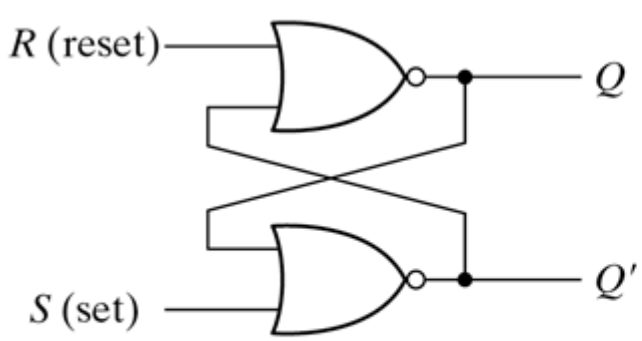


(b) Timing diagram of clock pulses

Fig. 5-2 Synchronous Clocked Sequential Circuit

SR Latch

1. The SR latch with two cross-coupled NOR gate is shown in next slide.
2. By setting S to 1 (R=0), the output Q will be 1 that putting the latch in the set state
3. If S goes to 0 (R=0), the circuit remains in set state (Q=1);
4. By setting R to 1 the circuit goes to reset state (Q=0);
5. The undefined state is when both input are 1



(a) Logic diagram

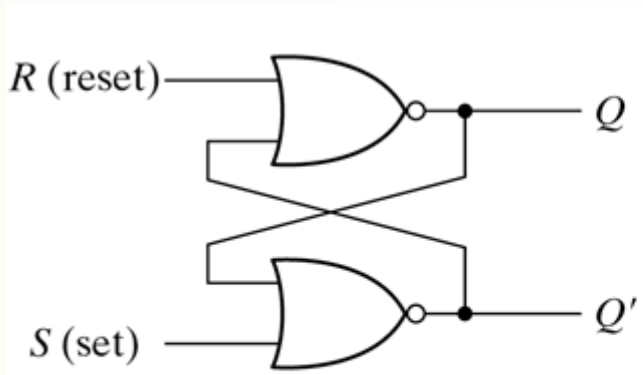
S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(undefined)

(b) Function table

SR Latch with NOR Gates

KENT STATE UNIVERSITY

SR Latch

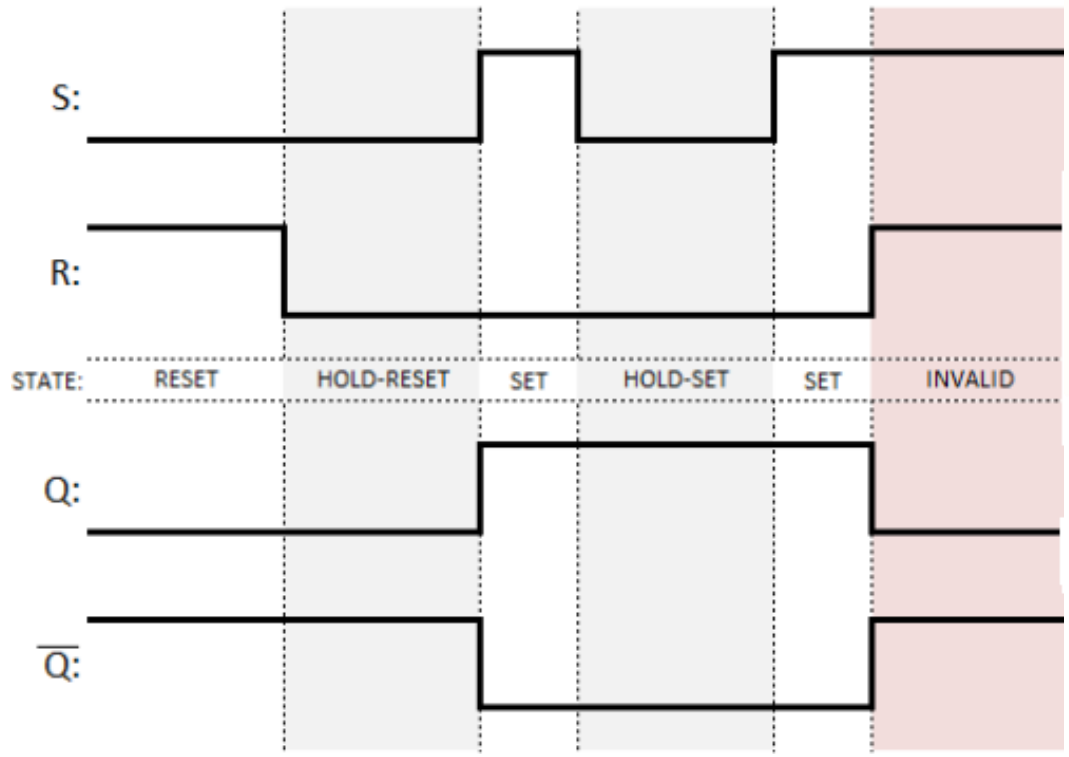


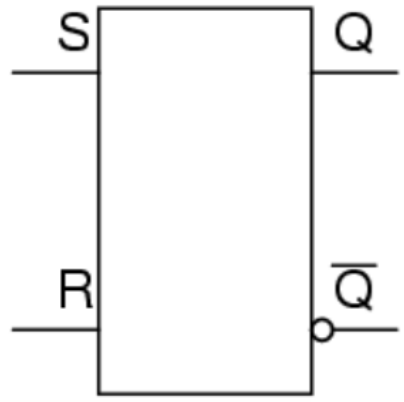
(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	

(b) Function table

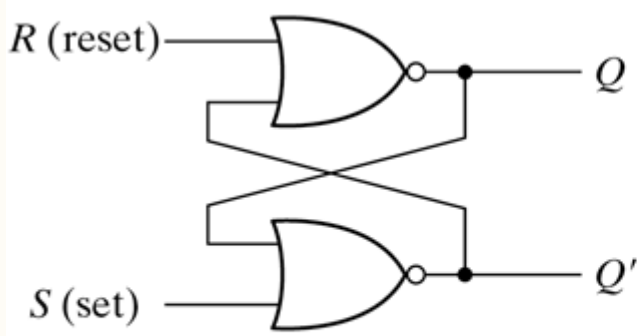
SR Latch with NOR Gates





KENT STATE UNIVERSITY

SR Latch

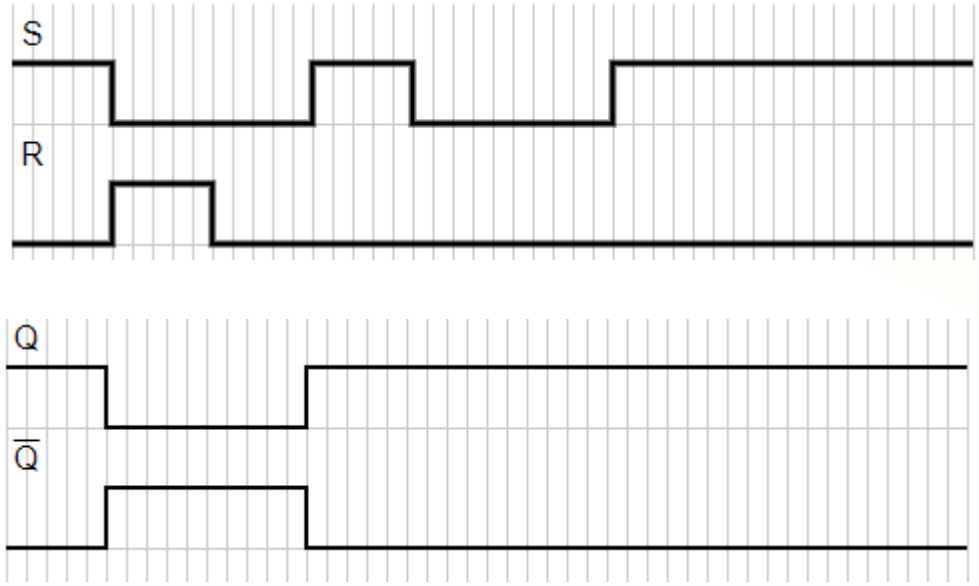


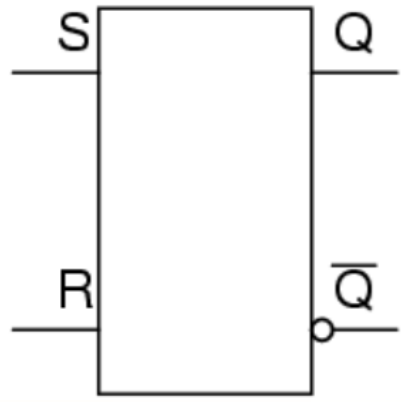
(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	

(b) Function table

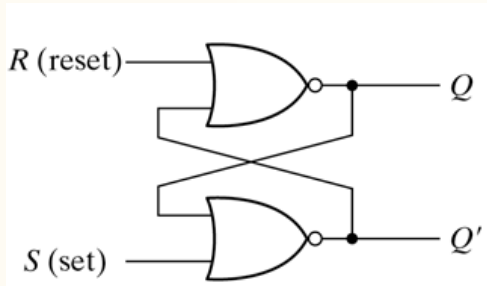
SR Latch with NOR Gates





KENT STATE UNIVERSITY

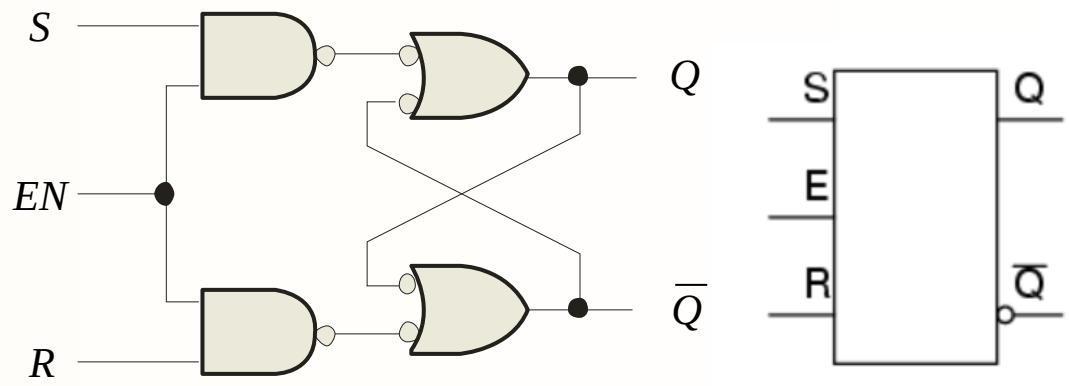
Gated SR Latch



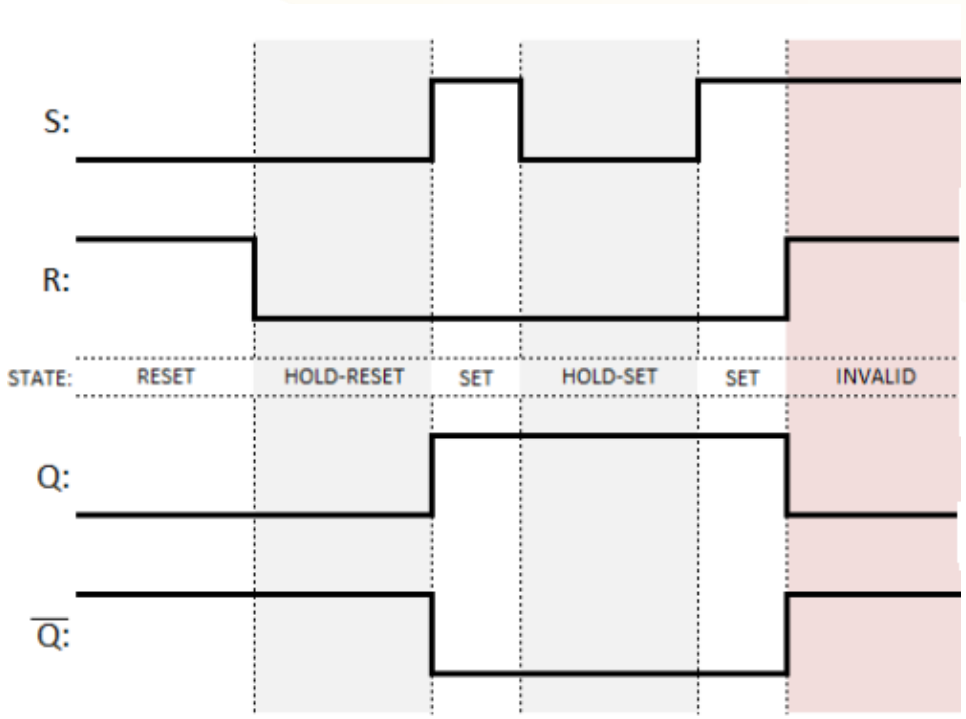
S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	

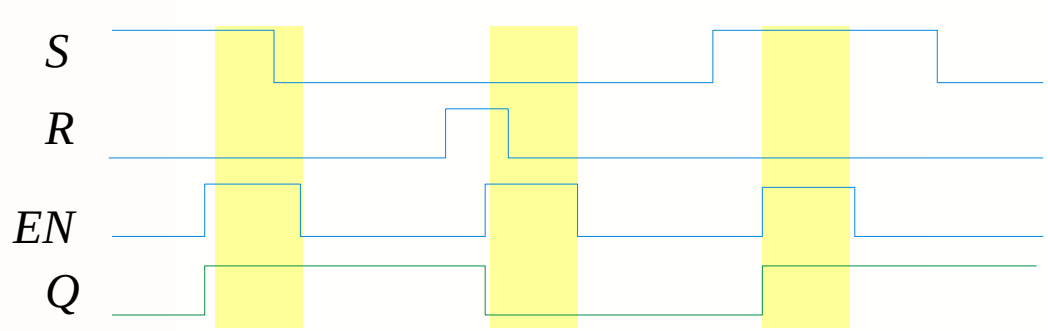
(a) Logic diagram (b) Function table

SR Latch with NOR Gates



Gated SR latch



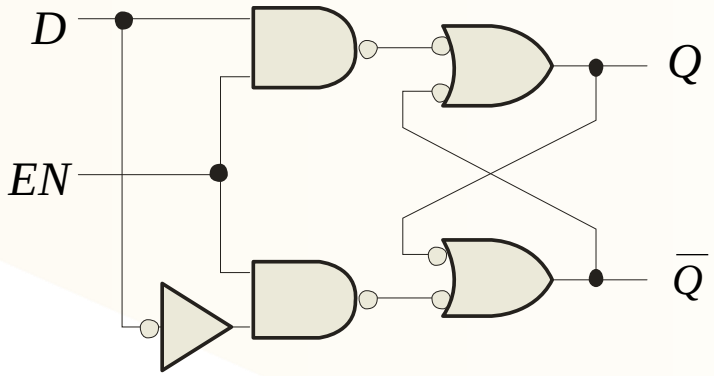


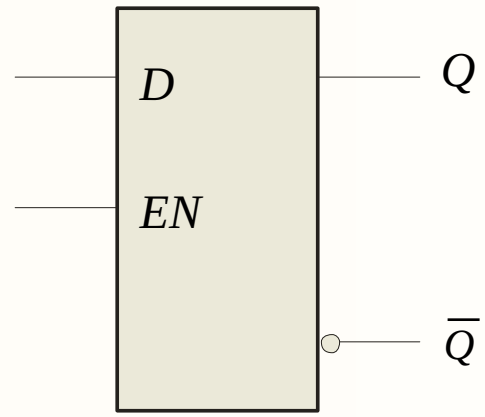
- The gated latch has an additional input, called enable (*EN*) that must be HIGH in order for the latch to respond to the *S* and *R* inputs.
- A gated latch is a variation on the basic latch.
- Keep in mind that *S* and *R* are only active when *EN* is HIGH.

KENT STATE UNIVERSITY

D Latch

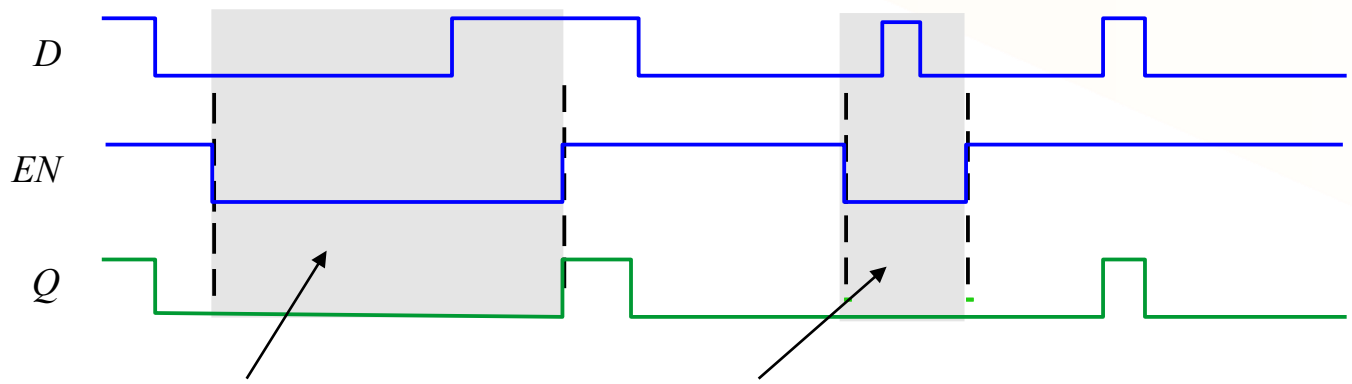
The *D* latch is a variation of the *S*-*R* latch but combines the *S* and *R* inputs into a single *D* input as shown:





A simple rule for the *D* latch is:

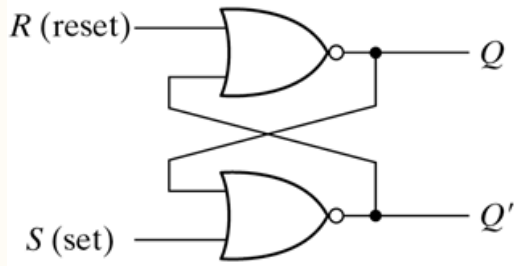
Q follows *D* when the Enable is active.



Notice that the Enable is not active during these times, so the output is latched.

KENT STATE UNIVERSITY

Practice



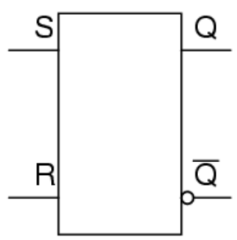
(a) Logic diagram

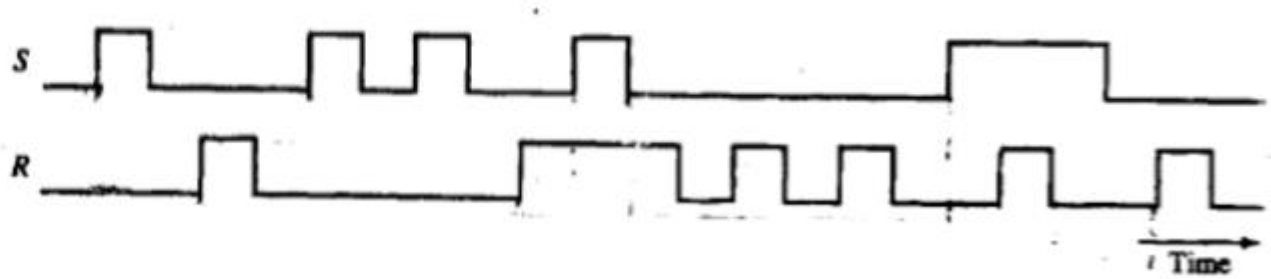
S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	

(b) Function table

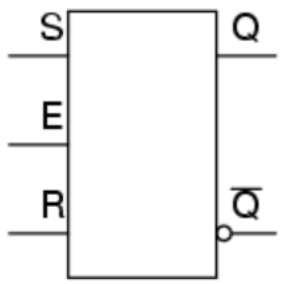
SR Latch with NOR Gates

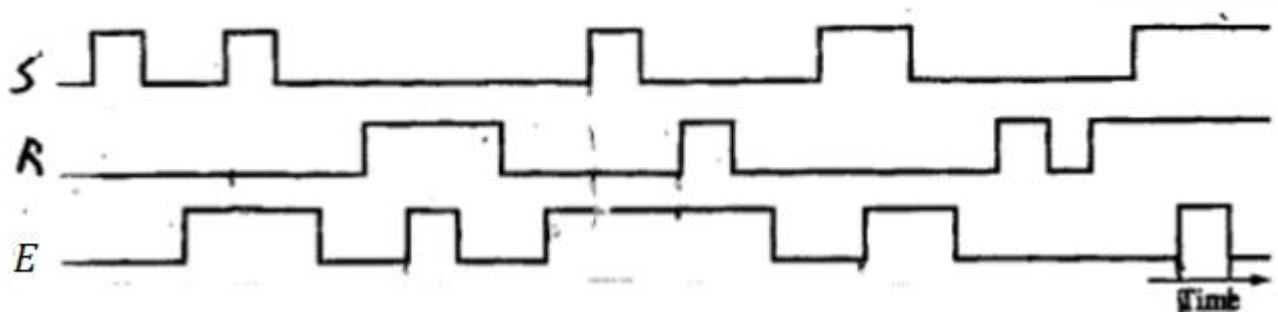
1) The input signals below are applied to the active high SR latch shown below, when Q is initially 0. Sketch the Q and Q' outputs.





(2) The input signals below are applied to a gated SR latch, when Q initially 0. Sketch the Q and Q'.





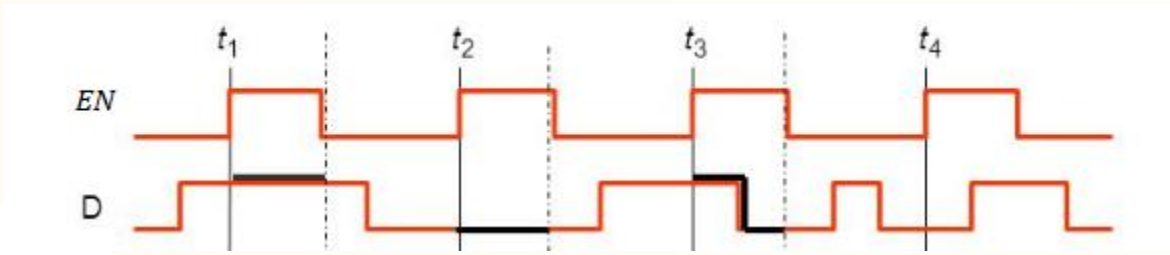
KENT STATE UNIVERSITY

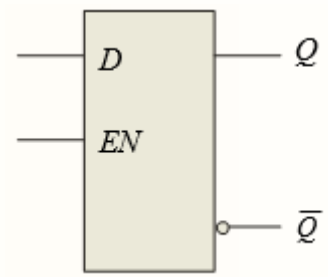
Practice

A simple rule for the *D* latch is:

Q follows *D* when the Enable is active.

(3) The input signals below are applied to a D latch, when Q initially 0. Sketch the Q and Q'.

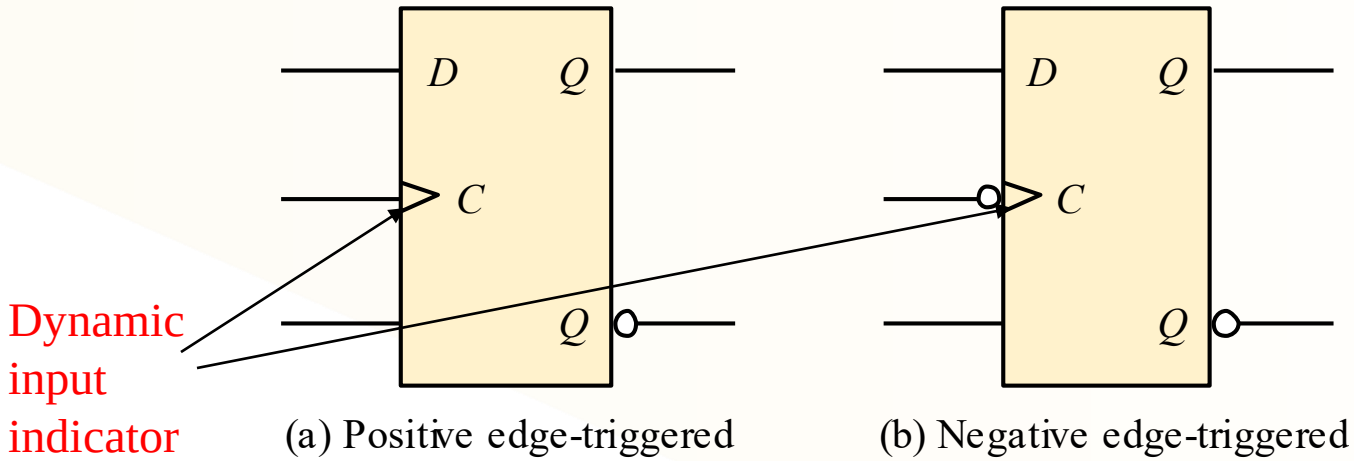




D Flip-Flops

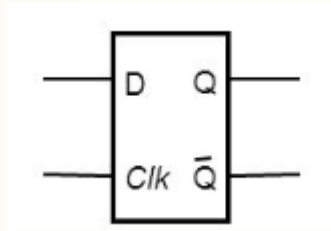
A flip-flop differs from a latch in the manner it changes states. A flip-flop is a clocked device, in which only the clock edge determines when a new bit is entered.

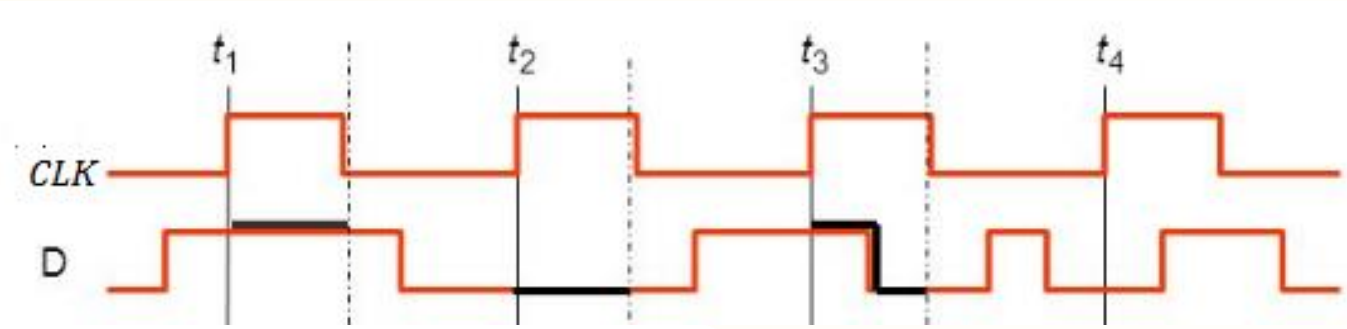
The active edge can be positive or negative.



D Flip-Flops

A flip-flop differs from a latch in the manner it changes states. A flip-flop is a clocked device, in which only the clock edge determines when a new bit is entered.



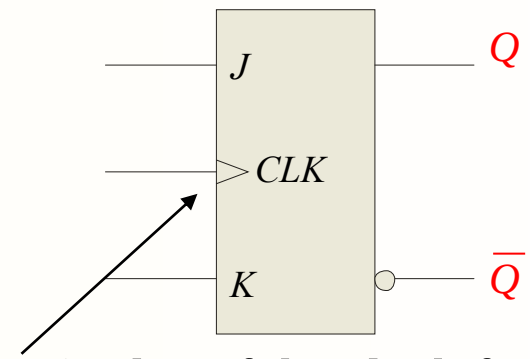


KENT STATE UNIVERSITY

J-K Flip-Flops

The J-K flip-flop is more versatile than the D flip flop. In addition to the clock input, it has two inputs, labeled *J* and *K*. When both *J* and *K* = 1, the output changes states (toggles) on the active clock edge (in this case, the rising edge).

Inputs			Outputs		
J	*K*	CLK	*Q*	$\overline{Q}$	Comments
0	0	↑	Q_0	$\overline{Q}_0$	No change
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	$\overline{Q}_0$	Q_0	Toggle



Notice that the outputs change on the leading(rising) edge of the clock for this J-K flip-flops.